ARO Planning Workshop
Security of Embedded Systems and Networks

John A. Stankovic
University of Virginia
January 25, 2005

***Position Statement***

Wireless sensor networks (WSNs) are composed of large numbers of minimal capacity sensing, computing, and communicating devices. These devices operate in complex and noisy environments. Transient and permanent random failures are commonplace. The considerable redundancy in such systems creates great potential for designing them to continue to provide their specified services even in the face of large numbers of such failures. WSNs are also susceptible to malicious, non-random *security* attacks. For example, a wireless sensor network deployed in remote regions to detect and classify targets could be rendered inoperative by various security attacks. To meet realistic requirements, WSNs must be able to continue to operate satisfactorily in the presence of, and to recover effectively from, security attacks. We propose that safe self-healing and adaptive infrastructures can work together to permit WSNs to continue to operate and self-heal in the presence of failures and security attacks.

***Key Limitations and Challenges***

1. Many current security solutions are heavyweight. Lightweight versions of current solutions and entirely new, but lightweight solutions are required for WSNs.
2. Many current solutions are not reactive; they attempt to provide security all-the-time. This is often not practical. We require new architectures and systems that can support dynamic reaction to attacks.
3. Detecting attacks is a serious problem in WSNs because of the limited resources that can be assigned to detection and because of the very noisy and failure prone devices, making it more difficult to distinguish between faults and security attacks. This includes stealthy denial of service attacks. The challenge is to solve these problems with new hardware and software solutions.

***Promising Innovations***

The confluence of four technologies is creating an opportunity for a major new approach to solving some aspects of the security problems for wireless sensor networks. Note that detecting attacks is not addressed in this position statement due to lack of space.

First, WSNs have now evolved to where there are large numbers of decentralized protocols integrated into functioning systems. However, these protocols have not typically addressed security. Large numbers of small sensor nodes each executing decentralized control protocols can provide a basis for new solutions that allow operation in the presence of attacks. For example, VigilNet, the system we developed under the Darpa NEST contract has well over 20 protocols operating in a decentralized manner and can form the basis for (initially) masking security threats; subsequently, based on concepts described below corrupted data or software components are self-healed. In other words, if designed with security in mind, decentralized protocols can prevent attacks in one part of the network from affecting the entire system. We consider the aggregate

behavior portrayed by large numbers of decentralized entities a critical technology, but it now needs to be "applied" to security problems.

Second, self-healing technology has progressed in the mainframe and Internet worlds to a degree where some of these ideas can now migrate to WSNs (subject to new sensor network constraints). Self-healing provides a means for masking and repairing security attacks. We suggest a need for extensions to self-healing technology that ascertain if the self-healing actions are safe (first off-line and then on-line) before they are placed into effect. The safety is ascertained by checking that dynamic integration of components to heal problems caused by security attacks meets required security, integrity and interface requirements. A key challenge is to ensure that the self-healing itself does not introduce new security vulnerabilities and make it easier to attack the system.

Third, emerging concepts in advanced aspect-oriented program design promise to allow for a separation of component, component-integration, security mechanism, and other concerns that, in turn, can enable important capabilities for the dependable design and dynamic adaptation of WSNs. Dynamic integration and re-configurability and the separate modularization of dynamically interchangeable security mechanisms and policy implementations are keys to enabling effective defense and self-repair. In particular, aspect-oriented separation of security-related code could facilitate the writing, verification, and updating of security code over time, allowing security countermeasures to evolve in the face of evolving threats.

Fourth, multi-hop wireless download can now support a real-time adaptive change to WSNs software. This can create diversity (downloading new components) and repair (downloading new allowable integration information), two concepts useful for addressing security attacks. Dynamically downloading and integrating updated security components is a key enabler for dynamic, evolving self-defense and repair under security attacks.

***Milestones***

1. Define new attack models for WSNs.
2. Create instances of lightweight security solutions for routing, localization, group sensor fusion, etc.
3. Develop hardware support for all aspects of security in WSNs.
4. Define a self-healing, security friendly architecture.
5. Implement and evaluate solutions in complete systems in realistic environments.

***Summary***

Wireless sensor networks have many uses for the military, e.g., they can provide surveillance in hostile areas, can help protect military installations, and can support urban warfare activities. The potential is unbounded. All of these capabilities will be jeopardized without built-in dependability and security capabilities. There is no way to finally solve the security problem as new attacks will always be conceived. We require solutions to enable WSNs to (i) operate in the presence of attacks, and (ii) evolve to support security changes needed over the lifetime of a system. The potential solutions mentioned here can help move WSNs from benign applications and environments to rugged and realistic systems that can achieve their true potential.

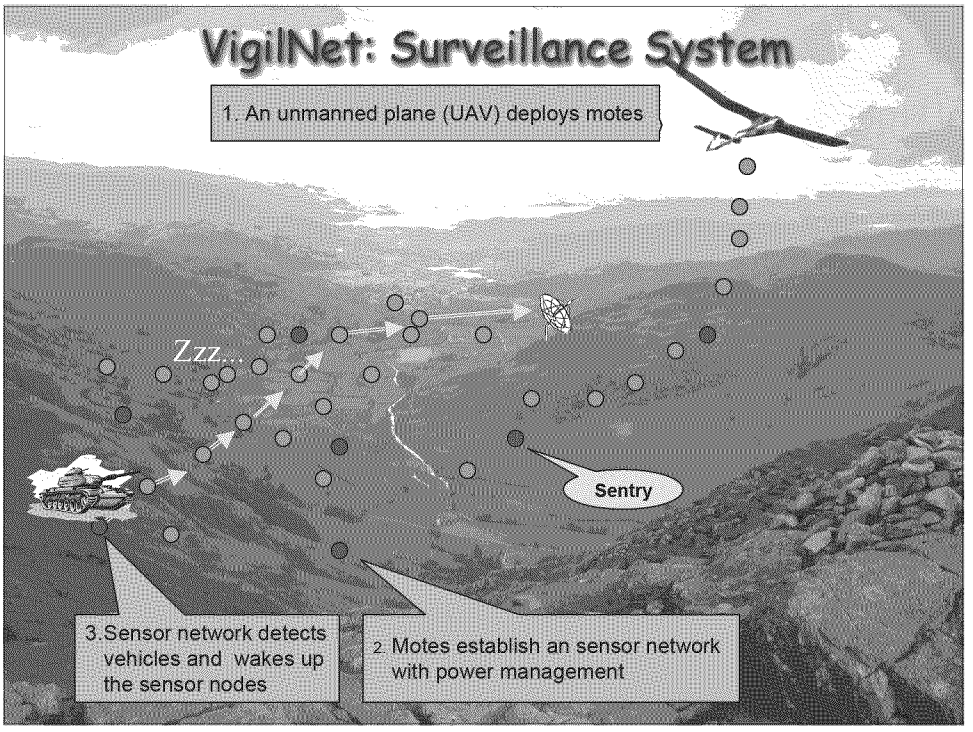
# Adaptable and Reactive Security for Wireless Sensor Networks

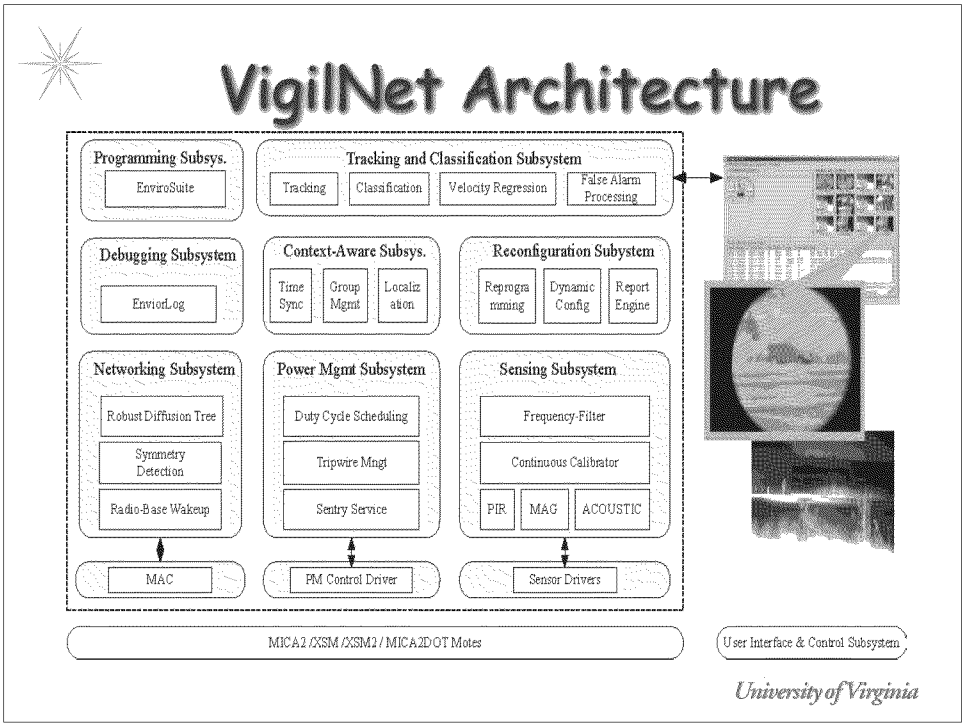
John A. Stankovic
Department of Computer Science
University of Virginia



## Outline

- Brief Motivation
- Adaptable Self-Healing Architecture
  - Components
  - AOP
  - Robust Decentralized Control
  - Lightweight Security Components
- Systems of Systems
- Summary


VigilNet: Surveillance System
1. An unmanned plane (UAV) deploys motes
Zzz...
Sentry
3.Sensor network detects vehicles and wakes up the sensor nodes
2. Motes establish an sensor network with power management


VigilNet Architecture
Programming Subsys.
EnviroSuite
Tracking and Classification Subsystem
Tracking
Classification
Velocity Regression
False Alarm Processing
Debugging Subsystem
EnviorLog
Context-Aware Subsys.
Time Sync
Group Mgmt
Localiz ation
Reconfiguration Subsystem
Reprogra mming
Dynamic Config
Report Engine
Networking Subsystem
Robust Diffusion Tree
Symmetry Detection
Radio-Base Wakeup
Power Mgmt Subsystem
Duty Cycle Scheduling
Tripwire Mngt
Sentry Service
Sensing Subsystem
Frequency-Filter
Continuous Calibrator
PIR
MAG
ACOUSTIC
MAC
PM Control Driver
Sensor Drivers
MICA2 /XSM /XSM2 / MICA2DOT Motes
User Interface & Control Subsystem
University of Virginia

# Security Issues

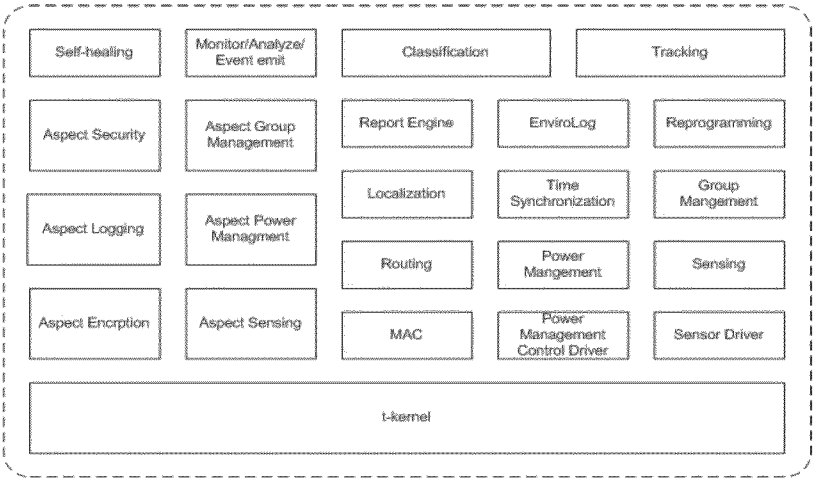
- Every one of the 30 services can be attacked
- Too expensive to make each service attack proof
- Attacks will evolve anyway



# Security Approach

- Operate in the presence security attacks
  - Robust decentralized control
- Self-Heal
  - AOP
- Evolve to new, unanticipated attacks
  - AOP and Wireless Downloads
- Lightweight solutions required due to severe constraints

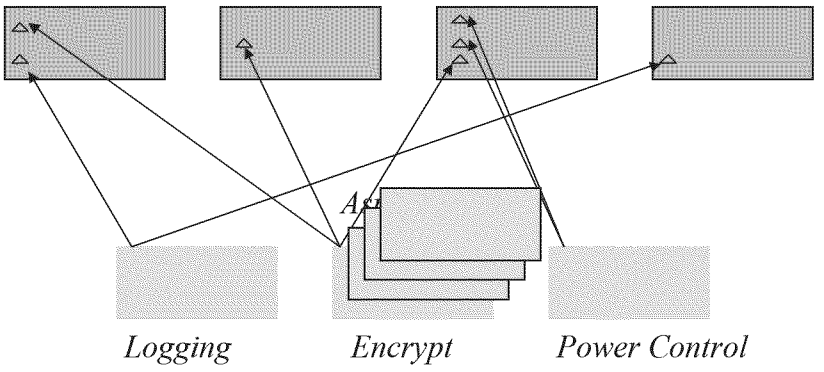
# Components






# Aspect Oriented Programming (AOP)

## Functional Modules



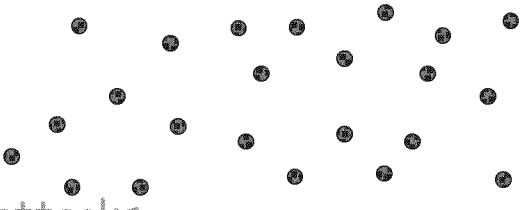
# Unanticipated Attacks

- What if advice was not available on the nodes
  - Typical for an unanticipated attack
  - Report event to base station
  - Find/Write new aspects
  - Disseminate to nodes



# Decentralized Control

- Large Numbers of Nodes
  - Aggregate Behavior Emerges
  - Control/Guarantee Behavior
- Redundancy
- Mask faults/ attacks
- Uniformity a problem/diversity



# Lightweight Components

- Secure (reactive/adaptive) routing
- Localization



# SIGF

- The SIGF family provides incremental steps between stateless and shared-state protocols.

| Protocol | General Approach | Corruption | Wormhole | HELLO flood | Black hole | Sybil | Replay DoS |
|---|---|---|---|---|---|---|---|
| IGF | Dynamic Binding | √ | √ | √ | -- | -- | -- |
| SIGF-0 | Nondeterminism | √ | √ | √ | √ | -- | -- |
| SIGF-1 | Local Reputation | √ | √ | √ | √ | √ | -- |
| SIGF-2 | Cryptography | √ | √ | √ | √ | √ | √ |

- SIGF allows efficient operation when no attacks are present, and good enough security when they are.

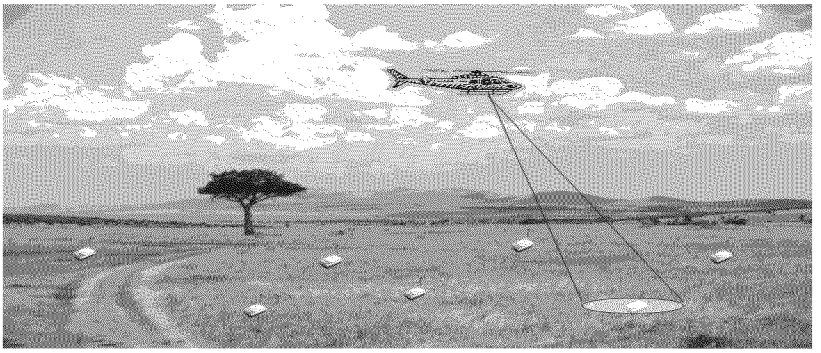
## Adaptive, Configurable

- Security level can be adaptive based on the resource constraints and security requirements.
- Each level can be configured based on parameters.



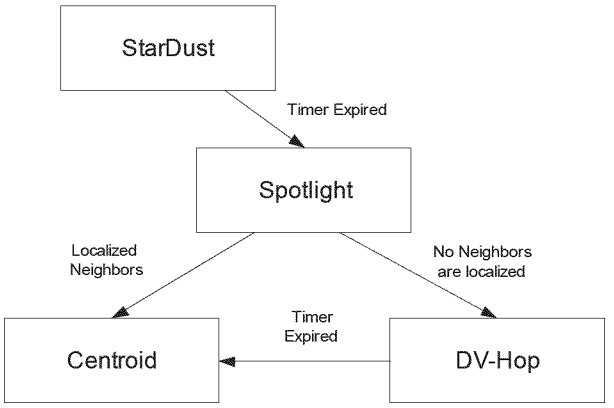
## Localization - Spotlight

- Run time-sync protocol
- Generate (invisible) light events
- Sensor nodes detect the events and report the timestamps
- The Spotlight device computes the location of the sensor nodes

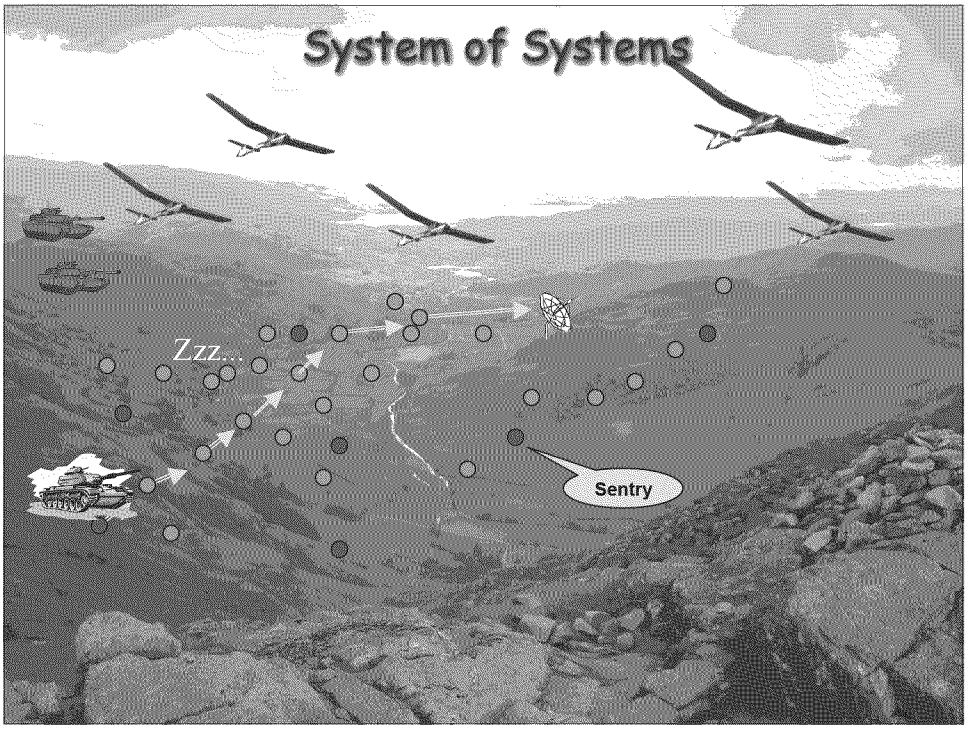


# Localization Robustness

- Execute combination of protocols




*University of Virginia*

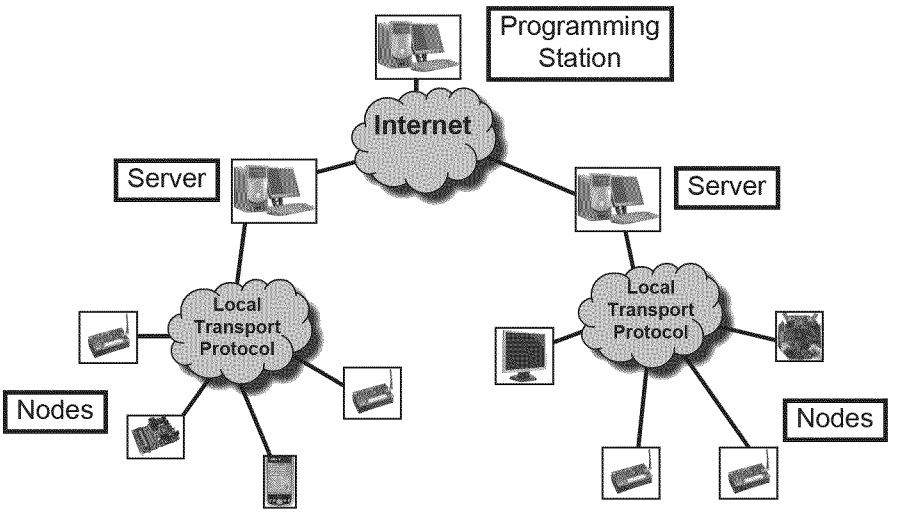


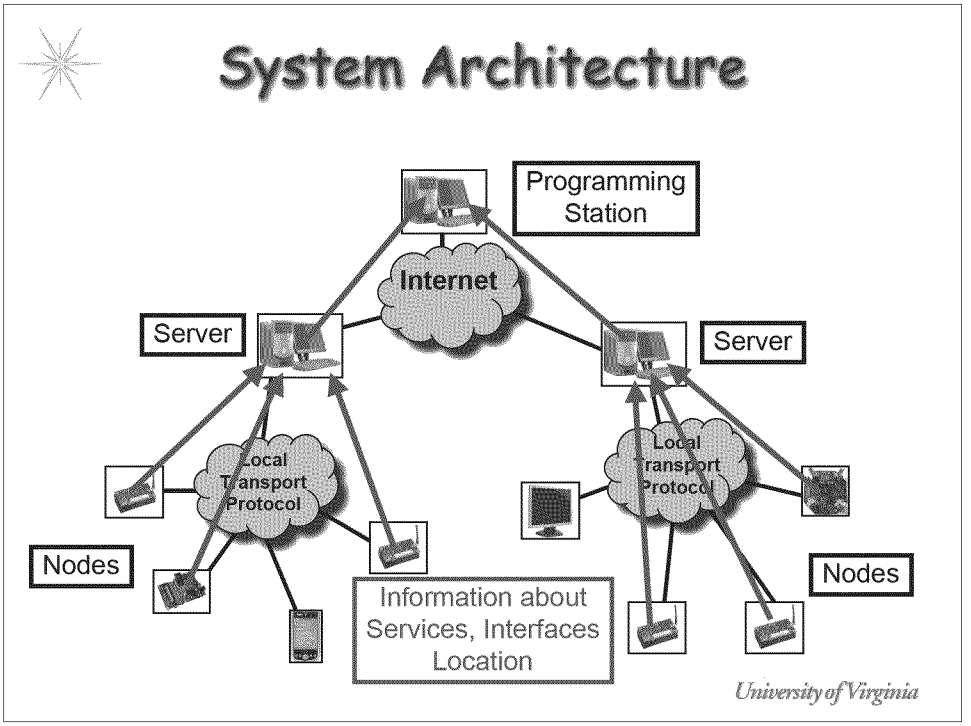
# Systems of Systems

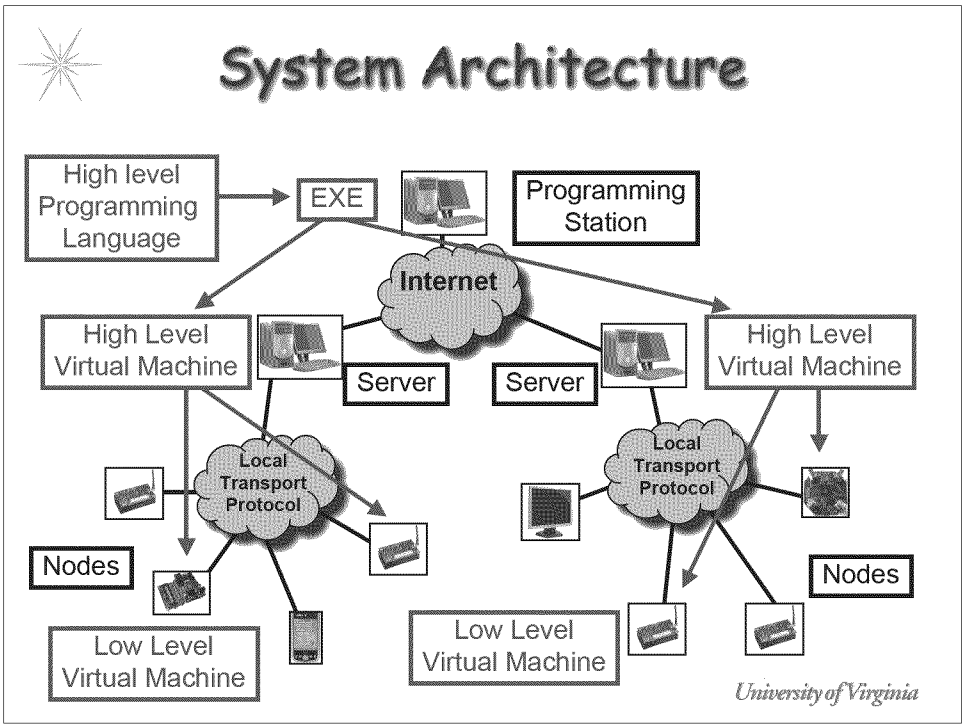
- Inter-system security
- How to program and debug to ensure
  - Behavior
  - Robustness



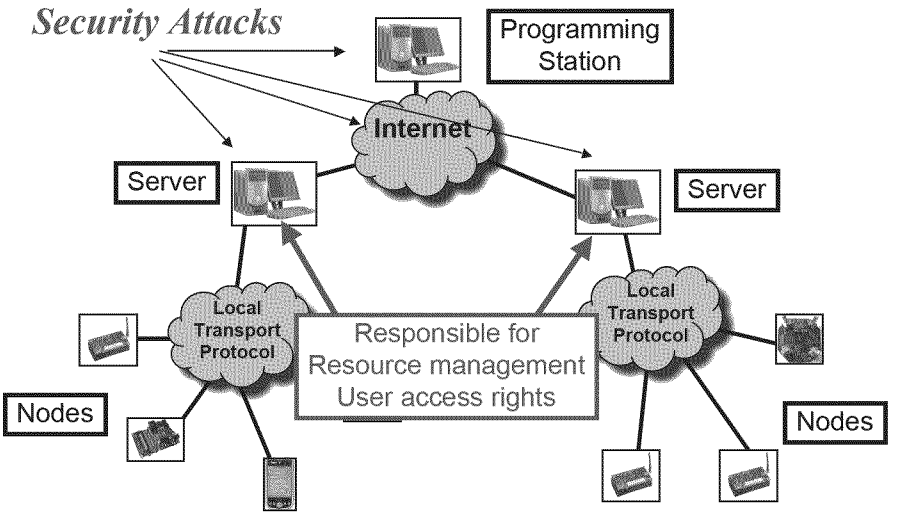
# System Architecture




System Architecture
Programming Station
Internet
Server
Server
Local Transport Protocol
Local Transport Protocol
Nodes
Nodes
Information about Services, Interfaces Location
University of Virginia


System Architecture
High level Programming Language
EXE
Programming Station
Internet
High Level Virtual Machine
Server
Server
High Level Virtual Machine
Local Transport Protocol
Local Transport Protocol
Nodes
Nodes
Low Level Virtual Machine
Low Level Virtual Machine
University of Virginia

# System Architecture






# Summary

- Security in WSN
  - Lightweight
  - Attack resilience
  - Evolve for unexpected attacks
- Security in Systems of Systems
- Self-Healing Architecture
  - AOP extensions
  - Adaptive and lightweight protocols

# Acknowledgements


- Anthony Wood
- Hua Cao
- Radu Stoleru